



```
-----  
1      SUBROUTINE SUB(A,N)  
2      INTEGER N  
3      REAL A(ABS(N))  
4      WRITE(*,*) A  
5      END SUBROUTINE  
-----
```

FIG. 1A  
PRIOR ART

```
-----  
1      SUBROUTINE SUB(A,N)  
2      INTEGER N  
        IF (N.GE.0) THEN          ! EXPANSION CODE  
            TMP = N                ! EXPANSION CODE  
        ELSE                       ! EXPANSION CODE  
            TMP = -N               ! EXPANSION CODE  
        END IF                    ! EXPANSION CODE  
3      REAL A(TMP)  
4      WRITE(*,*) A  
5      END SUBROUTINE  
-----
```

FIG. 1B  
PRIOR ART

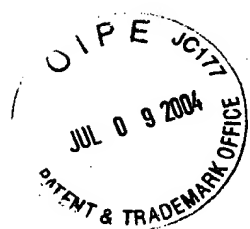


```
-----  
1      char *copy_string(char *s)  
2      {  
3          int i;  
4          char *buffer = (char*)malloc(strlen(s) + 1);  
5  
6          for (i = 0; s[i] != '\0'; ++i)  
7              buffer[i] = s[i];  
8  
9          return buffer;  
10     }  
-----
```

FIG. 2A  
PRIOR ART

```
-----  
1      char *copy_string(char *s)  
2      {  
3          int i;  
4          char *p; /* EXPANSION CODE */  
5          int tmp; /* EXPANSION CODE */  
6          tmp = 0; /* EXPANSION CODE */  
7          for (p = s; *p != '\0'; ++p) /* EXPANSION CODE */  
8              ++tmp; /* EXPANSION CODE */  
9          char *buffer = (char*)malloc(tmp + 1);  
10  
11         for (i = 0; s[i] != '\0'; ++i)  
12             buffer[i] = s[i];  
13  
14         return buffer;  
15     }  
-----
```

FIG. 2B  
PRIOR ART



```
-----
1      IF (Z.GT.EPS) THEN
2          A=B1
3      ELSE IF (ABS(Z).LE.EPS) THEN
4          A=B2
5      ELSE
6          A=B3
7      END IF
-----
```

FIG. 3A  
PRIOR ART

```
-----
1      IF (Z.GT.EPS) THEN
2          A=B1
3a     ELSE
        IF (Z.GE.0.0) THEN      ! EXPANSION CODE
            TMP = Z             ! EXPANSION CODE
        ELSE                    ! EXPANSION CODE
            TMP = -Z            ! EXPANSION CODE
        END IF                  ! EXPANSION CODE
3b     IF (TMP.LE.EPS) THEN
4         A=B2
5     ELSE
6         A=B3
3c     END IF
7     END IF
-----
```

FIG. 3B  
PRIOR ART